EN/LZT 108 4505 R1B

Mobile Phones R520, T20 and A2628 Developers' Guidelines **WAP Services**

ERICSSON

Ericsson is the leading provider in the new telecoms world, with communications solutions that combine telecom and datacom technologies with freedom of mobility for the user. With more than 100,000 employees in 140 countries, Ericsson simplifies communications for its customers - network operators, service providers, enterprises and consumers - the world over.

WAP™ and WAP Forum™ are registered trademarks of the WAP Forum, LTD., a registered company in the United Kingdom.

Revised edition (January 2001)
Publication number: EN/LZT 108 4505 R1B
This document is published by **Ericsson Mobile Communications AB**, without any warranty.
Improvements and changes to this text necessitated by typographical errors, inaccuracies of current information or improvements to programs and/or equipment, may be made by **Ericsson Mobile Communications AB** at any time and without notice. Such changes will, however, be incorporated into new editions of this document.
All rights reserved.
© Ericsson Mobile Communications AB, 2000

Contents

PREFACE 2

Who Should Read this Guide? 2

Overview of Contents 2

Related Publications 2

Ericsson Publications 2

Other Publications 3

Terminology 3

Abbreviations 3

Definitions 3

PHONE OVERVIEWS 5

Ericsson R520 5

Ericsson T20 5

Ericsson A2628 6

GENERAL USABILITY ASPECTS 7

USER INTERFACE ISSUES 8

The Browser Display 8

The Keypad 9

Navigating 10

DESIGN COMPONENTS 11

Using Text 11

Font Size and Font Style 11

Line Spacing and Line Breaks 12

Paragraphs 13

The Header 14

Using Select Lists 14

Single Choice Lists 15

Multiple Choice Lists 16

Using Input Fields 17

Using Images 18

Using Hyperlinks 20

Using Images as Hyperlinks 22

Using Tables 23

Adding items to the option menu 25

Miscellaneous Functions 26

WAP OVER DIFFERENT BEARERS 27

WAP over GPRS 27

WAP over GSM Data 27

WAP over SMS 27

THE PIZZA PENGUIN EXAMPLE 28

PREFACE

This document describes how to design WAP services for the Ericsson Mobile Phones R520, T20 and A2628.

Who Should Read this Guide?

This document is intended for developers of WAP services who want a deeper insight into designing applications for the Ericsson R520, T20 and A2628 WAP browsers.

It is assumed that the reader is familiar with the WAP concept and WML.

Overview of Contents

The chapters in this document are organized as follows:

Phone Overviews
: This chapter gives an overview of the supported functions in the Ericsson R520, T20 and A2628 phones.

General Usability Aspects
: This chapter gives an overview of issues to be taken into account when developing WAP services for the Ericsson R520, T20 or A2628 phones.

User Interface Issues
: This chapter gives a description of the physical and logical input/output devices of the R520, T20 and A2628. The chapter also describes the way these input/output devices are used together with WAP services.

Design Components
: This chapter contains information about graphical user interface components. The chapter also contains several examples of WML code as well as figures showing how these are presented in the browsers.

WAP over Different Bearers
: This chapter describes the characteristics of different bearers.

The Pizza Penguin Example
: This chapter contains a complete WAP application in WML code.

Related Publications

Ericsson Publications

[1] *Ericsson WAP Info*
http://www.ericsson.com/WAP/

[2] *WAP White Paper*
February 1999
AU-System AB
http://www.ericsson.com/WAP/what/wapwp2.pdf

Other Publications

[3] *Wireless Application Protocol Architecture Specification*
Version 30-Apr-1998
Wireless Application Protocol Forum, Ltd.
http://www.wapforum.org/what/technical.htm

[4] *WAP Wireless Markup Language Specification, Version 1.1*
Version 16-JUN-1999
Wireless Application Protocol Forum, Ltd.
http://www.wapforum.org/what/technical_1_1.htm

[5] *WAP Wireless Markup Language Specification, June 2000 release*
Version 19-FEB-2000
Wireless Application Protocol Forum, Ltd.
http://www.wapforum.org/what/technical.htm

Terminology

Abbreviations

GIF	Graphics Interchange Format
GPRS	General Packet Radio Services
HSCSD	High Speed Circuit Switched Data
HTTP	Hypertext Transfer Protocol
SMS	Short Message Service
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WAP™	Wireless Application Protocol
WBMP	WAP Bitmap
WML	Wireless Markup Language
WTLS	Wireless Transport Layer Security

Definitions

Bookmark	A named reference to a URL. Allows the user to keep track of favourite pages and then easily return to them later on.
Card	A single WML unit of navigation and user interface. May contain information to present to the user, instructions for gathering user input, etc.
Card Contents	Subject data stored or generated at an origin server. The card contents display text, images, hyperlinks and other controls supported by WML as a response to a user request. When a user performs an action in the card contents, the WAP browser responds according to the WML specification.
Card Title	An attribute in the WML Card element specifying advisory information about the card.
Current Card	The card whose contents are currently displayed in the browser.

Deck	A collection of WML cards.
Image	WBMP or GIF image contained in a Card.
Newcontext	An attribute in the Card element indicating that the current browser context should be re-initialized upon entry to this card. The re-initialization includes removal of all previously defined variables and a reset of the navigation history.
Push	A technology to push information to a mobile phone without it having been requested.
WAP Application	A collection of WML cards, with the newcontext attribute set in the entry card.
WAP Profile	A group of network and browser settings for different WAP usage scenarios.
WAP Service	A WAP application residing on a web site.

PHONE OVERVIEWS

Ericsson R520

The R520 supports the GSM 900, 1800 and 1900 bands. With the standard battery, talk time is 7 hrs 35 min and standby time 200 hrs. The phone is 130x50x16 mm in size and weighs 105 g.

The R520 also supports General Packet Radio Services (GPRS). This technology facilitates high-speed transmission of packet data, which ensures that the user remains continuously connected to the Internet.

The phone supports WAP version June 2000 and the digital signature technology necessary for secure e-commerce. The phone also supports WTLS class 1, 2 and 3. WTLS class 1 supports encoding, class 2 supports encoding and server authentication. WTLS class 3 supports encoding, server authentication and client authentication.

The R520 supports the Mobile electronic Transaction (MeT) initiative that enables safe electronic commerce.

It is possible to run WAP over several bearers such as GPRS, High Speed Circuit Switched Data (HSCSD), GSM data and SMS. HSCSD makes accesses over GSM approximately three times faster than ordinary GSM Data connections.

The phone supports the WAP push technologies.

The phone also supports sending and receiving e-mails. To make text input more convenient, the phone also supports predictive text input.

Ericsson T20

The T20 supports the GSM 900 and 1800 bands. With the standard battery, talk time is 10 hrs and standby time 200 hrs. The phone is 101x54x28 mm in size and weighs 128 g.

It is possible to run WAP over GSM Data and SMS.

The phone supports WAP 1.1 and additional functionality for the WAP mobile chat. To improve the use of chat functions, the phone also supports the Swatch Internet time concept.

Besides these functions, the T20 supports SMS templates in order to make SMS messaging simpler.

Ericsson A2628

The A2628 is a dual band phone supporting the GSM 900 and the GSM 1800 bands. With the standard battery, talk time is 6 hrs and standby time is 125 hrs. The phone is 131x51x25 mm in size and weighs 140 g.

The phone supports WAP over SMS and GSM Data.

WAP version 1.1 is supported in the A2628.

GENERAL USABILITY ASPECTS

Simplicity is the main key when designing WAP services. There are two reasons why it is important to keep the sites simple. The first reason is that the WAP standard is designed for use in devices with small displays, limited user input facilities and narrowband network connections. The second reason is that users of mobile devices are primarily interested in finding the information they want quickly. It is, therefor, important to sort out any irrelevant information and to minimize the path to the information.

There are several good tips on how to get a well-organized and easy to comprehend graphical interface. The first thing that has to be considered is the design and structure of the service. This means that the service should be divided into several smaller parts. Since overview is crucial, keep these parts relatively small. Do not force the user to scroll through more than 4-5 displays. Using links is an effective way of allowing the user to navigate through the different parts. In order to allow the user to find the right part easily, the hierarchy must not be too complex. Try to find a menu system that is neither too deep nor can be misinterpreted.

The size of the decks should be limited because of limitations in bandwidth and memory of the mobile device. This does not mean that decks should be as small as possible. Since all cards in a deck are loaded at the same time it is wise to put cards that often access each other in the same deck. The following table shows the maximum deck and image sizes.

Phone	Max deck size	Max image size	Max no. of images
R520			
T20	1600 bytes	1500 bytes	No limit
A2628			

Since it is relatively complicated to input data into a handheld device, it is important to make required inputs short and simple. One example of how the service designer can help the user is to pre-define the default values in select lists. Accesskeys are another aid that can simplify navigation.

In order to give the user a fast overview of the information, it is important to use headers. These headers are set to the title of the card. The title of the card should, therefor, say something about what can be found in the card. A concise design of all pages will also help the user to get a fast overview. In order to avoid unnecessary word wrapping, use short words instead of complicated ones.

Our best advice is to always think of the user and to test the pages on all phones.

Tip: *All Ericsson browsers can be identified at the server side by the contents of the HTTP/1.1 header User-Agent request field. At the server side, knowledge of the requesting browser's identity can then be used in order to start WAP services that are tailor-made for their particular browser.*

The above-mentioned identity can be found in the following table.

Phone	WAP version	Identity
R520	WAP June 2000	"EricssonR520/R1A"
T20	WAP 1.1	"EricssonT20/R1A"
A2628	WAP 1.1	"EricssonA2628/R2A"

The last part of the identity is the revision number and may be changed in future releases.

USER INTERFACE ISSUES

The Browser Display

All phones covered in this document have a graphical display. The size and resolution of these displays can be found in the table at the end of this page.

In most WAP phones the pixel width is not as large as the height. The pixel stretch is the ratio between the height and the width. This measurement is important to keep in mind when designing images.

The browser display has two sections: the Browser area and the Header at the top. Displays with four rows or more keep the header visible at all times. In addition to the browser area and header, dynamic overlays are used in connection with select lists and input fields.

Browser Area

The browser area is where the card content is presented to the user. The size of this area is dependent on the phone and the user set preferences.

If a card content is too extensive to be visible in full, only the beginning of the card will be visible when first displayed. By use of the navigation keys, the remainder of the card content can be viewed.

Text in the browser area is wrapped by default.

Graphical components, text and images are shown in the same order as they appear in the WML code

Phone	Resolution	Screen size	Colours	Pixel stretch	No. of lines
R520	101x67 pixels	30x23 mm	4 greyscales	1.24	4, 5, or 6
T20	101x33 pixels	32x12 mm	black/white	1.10	3
A2628	101x54 pixels	30x19 mm	black/white	1.20	4

Header

In the header, the uppermost line of the display, the identity of the Current Card is presented. If the card has a defined title this is shown. Otherwise the first characters of the URL are shown (excluding the “http://”). The small bold font is used in the header.

Tip: *Always use card titles and keep them short and meaningful.*

Overlays

Overlays are display windows that partly cover the browser area and header. The number of lines that can be shown in an overlay window is dependent on the phone and the user set preferences. Overlays are used in connection with select lists and in input fields. The user can scroll in an overlay window if it contains more lines than can be shown at a time.

The Keypad

The table below explains important key functions, used in connection with WAP Services.

Key	Normal Press	Long Press
YES	**YES** is used to select items in single choice lists, accept an entry in input fields and follow hyperlinks. The **Options menu** is brought up if no field is highlighted in the browser area.	Brings up the **Options menu** regardless of where the highlight is.
NO	Goes to the previous card. Cancels any select or input operation.	Exit to standby except when in an input field or a select operation
C	Erases anything entered in the input field, character by character.	Auto-repeats the erase operation.
▲ (UP)	Go to the previous line.	Auto-repeat normal press.
◄ (LEFT)	Navigates left, column by column, in tables. Used in multiple choice lists for checking or unchecking choices.	Auto-repeat normal press.
▼ (DOWN)	Go to the next line.	Auto-repeat normal press.
► (RIGHT)	Navigates right, column by column, in tables. Used in multiple choice lists for checking or unchecking choices.	Auto-repeat normal press.
◤ (UP+LEFT)	Go to previous line. Navigates sideways, column by column, in tables.	Auto-repeat normal press.
◢ (DOWN+RIGHT)	Go to next line. Navigates sideways, column by column, in tables.	Auto-repeat normal press.
▲ (Volume+)	Go to the previous line.	Auto-repeat normal press.
▼ (Volume-)	Go to the next line.	Auto-repeat normal press.
0 – 9	Used for entering numbers and characters.	
*	Used for changing between upper and lower case.	
#	Used in multiple choice lists for checking or unchecking choices.	
	Open the options menu.	

Note: *The Ericsson T20 and A2628 use the up-left and down-right buttons while the R520 uses the up, down, left and right buttons. The explicit option menu button is only supported by the R520.*

Navigating

In order to navigate to a WAP service, several methods are available.

- Going to a homepage. There can be one homepage for each WAP profile. See the User's Guide for the phone for details.
- Frequently visited locations can be bookmarked once and then reached easily using the **Bookmarks** submenu. When bookmarking, the title of the application entry card is the default name of the bookmark.
- New locations can be reached by directly entering the URL of the service, using the keypad. The **Enter Address** menu item is used in order to access this method.
- Services pushed to the phone can be found in the **Push Inbox**.
- Links received in SMS messages can be browsed.

Browsing through information in an application is done by scrolling using the directional arrow keys and by activating hyperlinks.

The directional arrow keys give a flexible and natural way to navigate between lines and links within a card. The horizontal navigation keys are used to facilitate a natural and logical navigation in tables and non-wrapped text. In all other cases vertical navigation is used. Note that some phones have combined arrows for up-left and down-right.

The up and down keys scroll one line at a time.

In cards that support accesskeys the user can navigate to the hyperlinks with numeric keys.

In order to navigate backwards, the **NO** key is used. The navigation backwards can be repeated until a Card with the `newcontext` attribute is reached.

The Options Menu

The **Options menu** contains a set of very useful functions. It is a partly context-dependent menu, which temporarily replaces the whole browser area contents when the **Option** key is pressed or when the **YES** key is pressed and no hyperlink, input field or select list is highlighted. If the **YES** key is pressed and held down for a longer time, the **Options menu** is invoked regardless of possible highlighted items.

In addition to context-dependent functions, the **Options menu** always contains useful browser functions: **Return to page, Bookmarks, Enter address, Send as link, Reload, Go to Menu, Exit, Status** and **Homepage.**

Note: *The status menu that shows the status of the current connection is only available in the R520. The option button is only supported by the R520.*

Options list elements (defined by WML `do` elements) are also mapped to entries in the **Options menu** (see also Adding items to the option menu).

DESIGN COMPONENTS

This chapter gives an overview of the graphical user interface components and descriptions of their corresponding WML elements. These components are used when designing WAP services.

The “Look and Feel” of the components is described and guidelines are made on component usage. The WML elements and their attributes are also listed and a brief description is given.

The WML element descriptions are supplemented with WML code as well as figures and show the way the elements are displayed and handled in the phones.

Note: *The description below does not include all possible WML elements and attributes, but all elements affecting the graphical user interface.*

Using Text

Generally, text is wrapped according to word boundaries. However, if a word is longer than half the line width and starts anywhere on the first half of the line, the word is split up and wrapped. Such a split wraps an extra character in order to make room for a hyphen, causing a minimum of two characters to appear on the new line. If such a long word starts in the second half of the display, the whole word is wrapped to the next line.

Font Size and Font Style

The phones covered in this document use a proportional font. A proportional font gives a dynamic and minimized width for each character, which improves readability of the displayed texts. This generally allows more characters to be displayed per line.

The used typeface name is “Ericsson Sans Serif” and two different font sizes are used: small and normal. Since the R520 has user adjustable line heights, the physical font size depends on the current settings. The following table shows the font sizes for different phone settings.

Phone	Line settings	Line height	Normal font	Small font
R520	4 lines	17 pixels	14 pixels	10 pixels
	5 lines	13 pixels	10 pixels	8 pixels
	6 lines	11 pixels	8 pixels	8 pixels
T20	3 lines	11 pixels	10 pixels	8 pixels
A2628	4 lines	13 pixels	10 pixels	8 pixels

The fonts can be shown in either regular or bold style.

WML

The Ericsson browser supports emphasis. The following table shows the supported emphasis and its effects.

Emphasis	Font
`Em`	Bold style
`Strong`	Bold style
`i`	Ignored
`b`	Bold style
`u`	Underlined
`big`	Normal font in bold style
`small`	Small font

Note: *Only the R520 support the underline element. Be careful when using the `u` element, because underlined text normally indicates a hyperlink.*

Note: *If a paragraph is centered or right aligned, a new emphasis element within a line will cause a line break.*

A text formatting example

This example shows all possible combinations of fonts and styles.

```
<p align="center">
   Normal, regular<br/>
   <b>Normal, bold</b><br/>
   <small>Small, regular<br/>
   <b>Small, bold</b></small>
</p>
```

Line Spacing and Line Breaks

The line spacing in the Ericsson browser is a single line. If a row is too long to fit horizontally in the browser area, the row will by default be wrapped onto multiple lines.

WML

The beginning of a new line is defined by a `br` element.

The `br` element affects all contents in the browser view. Text, select lists, input fields, hyperlinks and tables are positioned on a new line according to the current alignment. It can also be used inside a table cell (see Using Tables below).

Tip: *Do not use `br` elements within hyperlinks as this might impair readability.*

A line break example

The example below shows how the `br` element will affect the card content appearance.

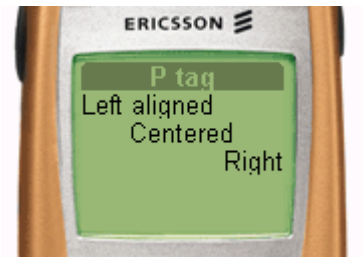
```
<wml>
<card id="init" title="BR tag">
<p>
   This text will wrap to a new line.
   <br/><br/>
   Preceded by 2 br tags.
</p>
</card>
</wml>
```

Paragraphs

Text on a WML card comprises one or more paragraphs.

A paragraph always starts on a new row.

Text within a paragraph is by default left aligned. However, other options are to display text right aligned or centered.



WML

Paragraphs are defined by using the `p` element.

The `p` element also establishes the alignment attribute for a paragraph.

Attribute	Description
`align`	The `align` attribute can take the following values: *left*, *right and center*. Left alignment is the default alignment mode.
`mode`	The `mode` attribute specifies the line wrap mode. The mode attribute can take the following values: *wrap* and *nowrap*. The default mode is *wrap*.

A paragraph example

The example below shows how the `p` element and `align` attribute affect the card content appearance.

```
<card id="init" title="P tag">
<p>
   Left aligned
</p>
<p align="center">
   Centered
</p>
<p align="right">
```

```
   Right
</p>
</card>
```

The Header

As mentioned earlier, the identity of the current card is presented in the uppermost line of the screen. See header on page 10.

WML

The card identity in the header is defined by using the `title` attribute in the `card` element.

Attribute	Description
`title`	The `title` attribute is shown as header of the card.

Using Select Lists

A Select list is an input component that specifies a list of options for the user to choose from. Single and multiple choice lists are supported. In a single choice list, the user may select one and only one item. In the multiple choice list, the user may select zero or more items.

A select list is accessed from the browser area through a select link, which is highlighted when in focus. The select link consists of a title and the currently selected values within angle brackets. In multiple choice lists the selected values are separated by commas.

If no item has previously been selected, a single choice list shows the first item and a multiple choice list only shows the brackets. By setting the `value` or `ivalue` attribute, this behaviour will be changed and the predefined items shown by default. See `value` and `ivalue` in the table below.

If the available space for the list of currently selected items is too small (smaller than the space needed for the text “WWW”) the list is wrapped to the row below the select title. Otherwise the values are truncated so that the list fits on the same line as the title.

When the user has opened the select list by pressing the **YES** key on the select link, the choice list is presented in an overlay window. The list items are presented as radio buttons in a single choice list and as check boxes in a multiple choice list. If a choice text preceding a radio button or a check box is too long, it is truncated.

The selection of items differs between single and multiple choice lists. In single choice lists, the user selects the desired item by pressing the **YES** key. When the user has made this selection, the select window is closed. In multiple choice lists, the user may select several items by checking their boxes. The boxes is checked or unchecked by pressing the left or right arrow or the **#** key. When the user has made the selections the select window may be closed by pressing the **YES** key. The user may always cancel the selections by pressing **NO**.

WML

Two elements define a choice list: `select` and `option`.

Multiple choice lists are specified using the `select` element with the `multiple` attribute set to *Yes*.

Each list item is specified by an `option` element.

The `select` element

Attribute	Description
`title`	The value of the `title` attribute is used both as select link text and as header in the overlay window.
`multiple`	The `multiple` attribute can take the following values: *Yes* and *No*. The default value is *No*.
`name`	The `name` attribute specifies the WML variable to set with the string value of the selected item. The `name` variable's value is used to pre-select options in the select list.
`value`	The `value` attribute specifies the default value when the name variable contains irrelevant data.
`iname`	The `iname` attribute specifies the WML variable to set with the index of the selected item. The `iname` variable's value is used to pre-select options in the select list.
`ivalue`	The `ivalue` attribute specifies the default item when the name variable contains irrelevant data.

Tip: *It is recommended to include the title attribute. However, if the title attribute is left out, the choice list will be headed "select" (translated into current display language).*

The `option` element

Text in the `option` element content is used as a description of the choice.

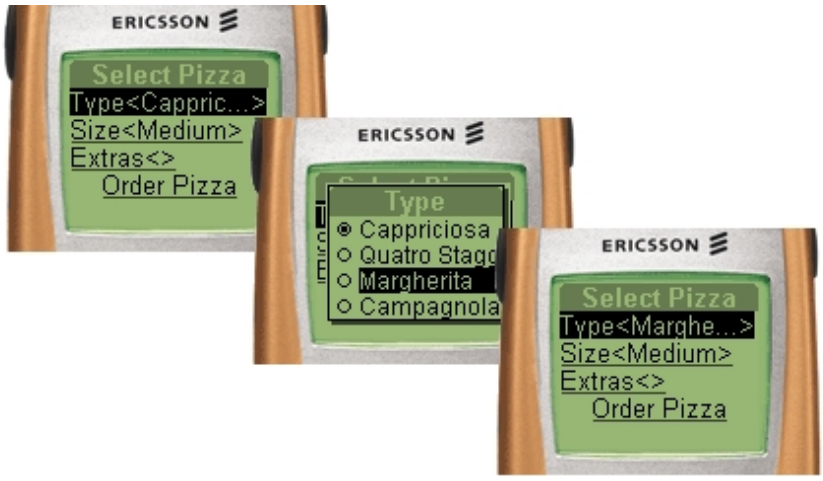
The `title` attribute is not supported for the `option` element.

The `optgroup` element

The `optgroup` element is used to group related option elements into a hierarchy. Elements grouped by the optgroup element will be rendered with the same tabulation.

Single Choice Lists

Single choice lists may be used to let the user select one item. The selections are presented as radio buttons in the overlay window.



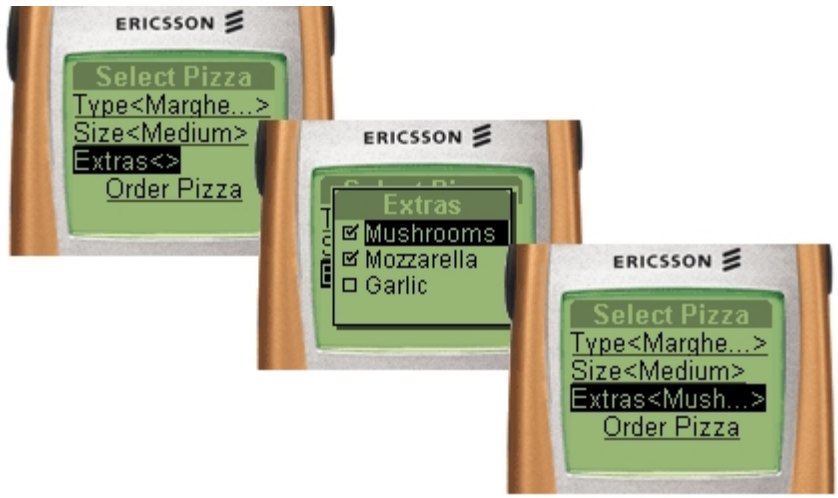
An example of a single choice list

```
<p>
   <select title="Type" name="Pizza">
      <option value="Capp">Cappriciosa</option>
      <option value="Quat">Quatro Staggioni</option>
      <option value="Marg">Margherita</option>
      <option value="Camp">Campagnola</option>
      <option value="Sici”>Siciliana</option>
   </select>
</p>

<p>
   <select title="Size" name="Size" value="M">
      <option value="L">Large</option>
      <option value="M">Medium</option>
      <option value="S">Small</option>
   </select>
</p>
```

Multiple Choice Lists

Multiple choice lists may be used to let the user select zero or more items. The selections are presented as check boxes.



An example of a multiple choice list

```
<p>
   <select title="Extras" iname="iExtra" multiple="true">
      <option>Mushrooms</option>
      <option>Mozzarella</option>
      <option>Garlic</option>
   </select>
</p>
```

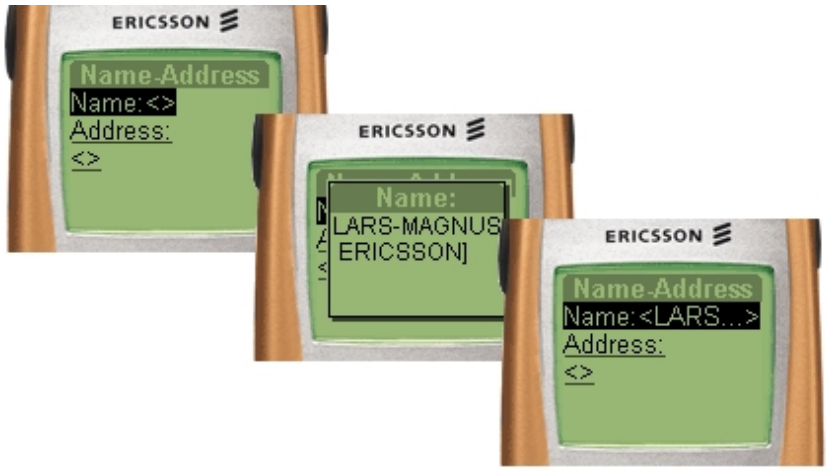
Using Input Fields

Input fields are the only components that enable a user to enter text.

Input fields are accessed from the browser area through an input link leading to an overlay window in which a new text can be entered or an already entered text can be edited. An input link is highlighted when in focus. After returning from the text input, by pressing **YES** or **NO,** the now current value of the text field is displayed within angled brackets.

If the available space after the input link is too small (smaller than the space needed for the text “WWW”) the text including brackets is wrapped to the next line. Otherwise the text value is truncated to fit on the same line as the input link. The input link remains highlighted.

The text in an input field is displayed using the normal bold font in the overlay window.



WML

Input fields are defined by using the `input` element.

The following attributes are supported:

Attribute	Description
`title`	The value of the `title` attribute is used both as input link text and as header in the overlay window.
`type`	The `type` attribute can take the following values: *text* and *password*. If `type` is set to *password*, asterisks will replace the characters.

`format`	Specifies an input mask for user input entries. The string consists of mask control characters and static text that is displayed in the input area. The format control characters specify the data format expected to be entered by the user. The format codes are: **A** Upper case alphabetic or punctuation **a** Lower case alphabetic or punctuation **N** Digits **X** Upper case alphabetic character, numeric character, symbol or punctuation character **x** Lower case alphabetic character, numeric character, symbol or punctuation character **M** Any character **m** Any character ***f** Any number of characters of type f **nf** n number of characters of type f **\c** Display any character c, in the entry field
`name`	The `name` attribute specifies the WML variable to set with the result of the user's text input.
`value`	If no pre-load value is defined for the input object, the value of the `value` attribute is used as a default text in the input field.
`emptyok`	The `emptyok` attribute specifies if empty inputs are accepted. The default value is dependent of the format mask. If the format mask allows empty inputs the default value is true otherwise false.

Tip: *It is recommended to include the title attribute.*

An input example

```
<p>
   <input type="text" title="Name:" name="Name" format="*A"/>
   <input type="text" title="Address:" name="Address"
   format="*M"/>
</p>
```

Using Images

The Ericsson browser supports images in WAP bitmap (WBMP) and GIF formats. The maximum image size and the maximum number of images can be found in the table on page 8.

An image is always loaded after the WML code. During the load time, the browser displays a rectangular placeholder with an optional text. The size of the placeholder depends on the length of the optional text. See the figure below.

Note: *The R520 supports 4-grayscale GIF images while T20 and A2628 supports black and white GIF images.*

However, the phone user can choose not to load images at all. In this case, placeholders become permanent.

A margin of 4 pixels at the top and 4 pixels at the bottom is added to the image space. An image is displayed on separate rows and uses the minimum number of text rows needed to fulfil the height of an image.

Images are shown centred horizontally and centred vertically.

There is no limitation to the height of an image. An image that is higher than the display can be viewed by scrolling up and down one row at a time. However it is recommended to use images not higher then the display height. The maximum image height to be fitted on one screen without the need to scroll is dependent on the phone and the current settings.

Phone	Nbr rows	Max image height
R520	4 rows	47 pixels
	5 rows	48 pixels
	6 rows	51 pixels
T20	3 rows	33 pixels
A2628	4 rows	35 pixels

If an image is wider than the browser area, it is centered and truncated at both ends.

WML

Images are defined by the `img` element.

The Ericsson browser supports the following attributes:

Attribute	Description
`alt`	An alternative textual representation for the image used in the placeholder.
`src`	The source (URI) of the image.

An image example

```
<p align="center">
   <img alt="baker"src="baker.gif"/>
   The oven is turned on and your pizza will be delivered just
in time.
</p>
```

Using Hyperlinks

The concept of hyperlinks is well known from the Internet environment. All hyperlinks in the Ericsson browser are underlined to mark that they are hyperlinks. When a hyperlink is in focus it is highlighted.

Some examples of hyperlink usage:

- Navigate within a deck
- Navigate to a new deck
- Start composing an SMS
  The Send New function is started and the e-mail address from the mailto URL is copied into the message area.
- Download a new melody
  The user is prompted to play and save the new melody.
- Download an electronic business card
  The user is prompted to save the business card.
- Download calendar data
  The user is prompted to save the calendar data.
- Download a certificate
  The user is prompted to save the certificate.
- Initiate a voice call
- Add telephone number to phonebook
- Send DTMF tones

A hyperlink may be located anywhere in the text flow. Text in hyperlinks is displayed in the same font as the text preceding the link, but the text is underlined. Selecting the link invokes the action associated to the link.

A helpful tool when using hyperlinks is the accesskey attribute. This attribute allows the user to select a hyperlink by pressing the defined button anywhere in the page. Accesskeys can be used in a, anchor and input tags. All elements that are associated with an accesskey are marked with a identifying icon in front of the actual tag. Notice that the browser may use different accesskeys other than those specified in the WML code.

WML

Hyperlinks are specified using the `a` element.

Attribute	Description
`href`	The `href` attribute specifies the destination of the URI.
`title`	The `title` attribute specifies a brief text string to identify the link.
`accesskey`	The `accesskey` attribute specifies an `accesskey` to the link.

Note: *The accesskey attribute is only available in the R520.*

Below are examples of useful functions in the Ericsson browser that can be activated through hyperlinks.

Download a new melody

```
<a href="Bamse.mel"> Bamse </a>
```

The file must have a `.mel` extension or the content type must be set to `text/x-vmel` on the server.

See Melody Composer found at Ericsson Mobile Internet for details on how to compose and store melodies.

Download an electronic business card

```
<a href="LarsMagnusEricsson.vcf"> Download Lars-Magnus Ericssons Business Card </a>
```

The file must have a `.vcf` extension or the content type must be set to `text/x-vcard` on the server.

Note: *Electronic business cards are only supported by the R520.*

Download calendar data

```
<a href="BoardMeeting.vcs"> Book board meeting in calendar</a>
```

The file must have a `.vcs` extension or the content type must be set to `text/x-vCalendar` on the server.

Note: *Calendar data are only supported by the R520.*

Download an certificate

```
<a href="EricssonCertificate.crt"> Download the Ericsson certificate</a>
```

The file must have a `.crt` extension or the content type must be set to `application/vnd.wap.wtls-ca-certificate` on the server.

Note: *Certificates are only supported by the R520.*

Start compose an SMS

```
<a href="mailto:pizza@penguin.nu">Send an SMS to Pizza Penguin</a>
```

Initiate voice call

```
<a href="wtai://wp/mc;+464690510">Call Pizza Penguin</a>
```

Initiates a voice call to the specified number. The user can see the number and is told how to acknowledge the call.

Add to phonebook

```
<a href="wtai://wp/ap;+464690510;PizzaPenguin">Add Pizza Penguin
to Phonebook</a> <br/>
```

Add the specified number to the phonebook.

Note: *Add to phonebook is only supported by the R520.*

Send DTMF tones

```
<a href="wtai://wp/sd;9">Next message</a>
```

Sends the specified DTMF tone sequence (i.e. the digit 9 in the example above) through an active voice connection. A voice call must have been initiated before the DTMF tones can be sent.

A hyperlink example

Three different links are shown in the example: a mailto link, a voice call link and an interdeck link.

```
<p align="center">
   <br/><a href="mailto:pizza@penguin.nu?subject=Pizza"
      accesskey="0">
   Send mail</a>
   <br/><a href="wtai://wp/mc;+464690510" accesskey="1">
   Call</a>
   <br/><a href="#Start">[Go to Start]</a>
</p>
```

Using Images as Hyperlinks

An image can also serve as a hyperlink. Images that are used as hyperlinks are marked with a 2-pixels wide frame around the image. The space for the frame is taken from the image space. This implies that images that are used as hyperlinks should not be too small.

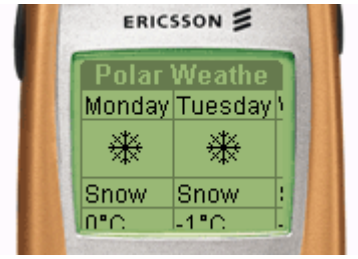
WML

An image hyperlink example

```
<p align="center">
   <br/>
   <a href="mailto:pizza@penguin.nu">
   <img alt="mail"src="mail.gif"/> </a>
   <a href="wtai://wp/mc;+464690510">
   <img alt="phone"src="phone.gif"/> </a>
</p>
```

Using Tables

Due to its large display and four-directional navigation possibilities, the Ericsson browser has good support for tables. Tables are used to make information easier to grasp and analyze.



The Ericsson browser displays a table in columns and rows with solid borders. A table is located on a new line. Empty table cells and empty table rows are significant and are included in the presentation of the table.

When a table is in focus it can be panned, column by column, using the horizontal navigation keys in order to view all columns.

The minimum column width is 20 pixels. The maximum number of rows is 90 and the maximum number of columns is 90. Rows and columns exceeding these maximum values are discarded. Note that the width of all columns containing an empty cell will be set to 90% of the screen width.

Text in a table cell can be left aligned, centre aligned or right aligned. All text-formatting rules apply to text within a table. The browser will handle a multiple-lines cell, which may result from wrapping of cell contents.

Tables are presented column by column on the screen. Each column width is calculated from the widest single cell up to a maximum of 90% of the display width, i.e. 91 pixels. Text content that is wider than this is wrapped to fit the display. This wrapping tries to put as much text as possible in a row and therefore wraps all characters, plus one to fit a hyphen, to the next row.

Tables can also be used to format card contents that include images in an appealing way. If, for example, you would like to have an image followed by a text on the same row, the image is put in the first column and the text in the second.

When using images in tables, a margin of 4 pixels at the top and 4 pixels at the bottom is added to the image space. If an image is wider than 90% of the cell width, the part of the image that is larger than 90% is truncated. There is no principal height limitation to an image.

In addition to text and images, tables can also contain hyperlinks and images used as hyperlinks.

WML

The `table` element is used together with `tr` and `td` to create a set of aligned columns of text and images.

The `table` element

Attribute	Description
`columns`	Specifies the number of columns for the table.

The `tr` and `td` elements

The `tr` element is used as a container to hold a single table row.

The `td` element is used as a container to hold single table cell data.

The Polar Weather `table` example

```
<card id="Weather" title="Polar Weather" newcontext="true">

<p>
   <table align="ccccc" columns="5">
   <tr>
      <td>Monday</td>
      <td>Tuesday</td>
      <td>Wednesday</td>
      <td>Thursday</td>
      <td>Friday</td>
   </tr>
   <tr>
      <td><img alt="weather" src="snow.gif" width="17"
         height="17" /></td>
      <td><img alt="weather" src="snow.gif" width="17"
         height="17" /></td>
      <td><img alt="weather" src="sun.gif" width="17"
         height="17" /></td>
      <td><img alt="weather" src="sun.gif" width="17"
         height="17" /></td>
      <td><img alt="weather" src="sun_in_cloud.gif" width="17"
         height="17" /></td>
   </tr>
   <tr>
      <td>Snow</td>
      <td>Snow</td>
      <td>Sunny</td>
      <td>Sunny</td>
      <td>Partly Cloudy</td>
   </tr>
   <tr>
      <td>0&#176;C</td> <!--&#176; = character for degrees-->
      <td>-1&#176;C</td>
      <td>-5&#176;C</td>
      <td>-5&#176;C</td>
      <td>-1&#176;C</td>
   </tr>
   </table>
</p>
</card>
```

Adding items to the option menu

Option items are components that start actions or change states.

All option items are mapped to entries in the **Options menu**. These dynamic menu items precede the static functions in the **Options menu**.

Note: Use hyperlinks instead of option items since hyperlinks are more accessible in the user interface.

WML

Option items are defined by using the WML `do` element.

Selecting the associated menu item in the **Options menu** invokes the action defined in the `do` element.

The Ericsson browser supports the following attributes:

Attribute	Description
`label`	The value of the `label` attribute is used as text in the Options menu. The text is not truncated.
`optional`	The `optional` attribute specifies if the user agent may ignore the `do` element.

Tip: *The option items are ordered by their type attributes in the options menu. An option item with the `type` accept is placed first, followed by an option item with the `type` prev. Following these, all other option items are placed in the order they appear in the card.*

Note: *On the T20 and the A2628 two do elements may not have identical `type` values, unless the `name` attribute is specified.*

An Option List example

```
<p align = "center">
   <do type="accept" label="What's On">
      <go href="#PizzaInfo"/>
   </do>
   .
   .
</p>
```

Miscellaneous Functions

Start WAP Services from an SMS

If an SMS message contains the URL of a WAP service, the user can select the URL, press **YES** and the URL is automatically copied into the **Enter Address** function. If the user presses **YES** again the browser is started and the WAP service loaded.

Send as link

The **Option menu** contains the function **Send as link** that allows the user to send the URL of the current card in an SMS to another phone.

WAP OVER DIFFERENT BEARERS

WAP over GPRS

By using GPRS the phone may be constantly connected to the Internet and all data is sent in packages.

Since GPRS has a comparably higher transmission speed than GSM Data and SMS, it is more suitable for large data transfers.

WAP over GPRS is suitable for all types of services. The packet-switched characteristics make GPRS ideal for pull services such as getting stock quotes and messaging and for push initiations.

WAP over GSM Data

When using WAP over GSM Data a simple circuit switched data call is established between the phone and the internet provider. This data call is maintained during the entire WAP session. The user pays for the duration of the data call.

The main advantage with GSM Data as a bearer is that it is supported by almost all operators.

WAP over SMS

When using WAP over SMS the point-to-point short messages services is used. All communication takes place with SMS messages between the phone and the SMS service centre. The user starts the communication by sending a WML request within an SMS message. The service centre responds by sending the requested page within one or several SMS messages.

The user only pays for the request message sent to the service centre; the responding messages are free. It is, therefore, advisable to avoid browsing among different decks while using SMS as a bearer and to avoid overlarge decks being sent over SMS. The reason for this is that each SMS message has a limited size and the probability of data loss will increase.

WAP over SMS is suitable for messaging services, information services and push initiations.

THE PIZZA PENGUIN EXAMPLE

Below is a complete WML code listing for a WAP demonstration application. Most of the examples in the previous chapters are selected from this application.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>

<card id="Start" title="Pizza Penguin" newcontext="true">
<onevent type="ontimer">
   <go href="#Main"/>
</onevent>
<timer value="70"/>
<p>
   <img alt="Penguin" src="mpenguin.gif"/>
</p>
</card>

<card id="Main" title="Select Pizza">
<p>
   <select title="Type" name="Pizza">
      <option value="Capp">Cappriciosa</option>
      <option value="Quat">Quatro Staggioni</option>
      <option value="Marg">Margherita</option>
      <option value="Camp">Campagnola</option>
      <option value="Sici">Siciliana</option>
   </select>
</p>
<p>
   <select title="Size" name="Size" value="M">
      <option value="L">Large</option>
      <option value="M">Medium</option>
      <option value="S">Small</option>
   </select>
</p>
<p>
   <select title="Extras" iname="iExtra" multiple="true">
      <option>Mushrooms</option>
      <option>Mozzarella</option>
      <option>Garlic</option>
   </select>
</p>
<p align = "center">
   <do type="accept" label="What's On">
      <go href="#PizzaInfo"/>
   </do>
   <a href="#Order">Continue</a><br/>
   <a href="#Contact">[Contact Us]</a><br/>
   <a href="#Start">[Go to Start]</a>
</p>
</card>
```

```
<card id="Order" title="Name-Address">
<p>
   <input type="text" title="Name:" name="Name"
      format="*A"/>
   <input type="text" title="Address:" name="Address"
      format="*M"/>
</p>
<p align="center">
   <br/>
   <a href="mailto:pizza@penguin.nu?subject=Pizza?body='Pizza:
      $(Pizza),Size:$(Size),Extras:$(iExtra),Name:$(Name),
      Adress: $(Address)'">Submit order</a><br/>
   <a href="#Contact">[Contact Us]</a><br/>
   <a href="#Start">[Go to Start]</a>
</p>
</card>

<card id="Contact" title="Contact Us">
<p align="center">
   <br/>
   <a href="mailto:pizza@penguin.nu">
   <img alt="mail" src="mail.gif"/> </a>
   <a href="wtai://wp/mc;+464690510">
   <img alt="phone" src="phone.gif"/> </a>
</p>
<p align="center">
   <a href="#Start">[Go to Start]</a>
</p>
</card>

<card id="PizzaInfo" title="What's On">
<p>
   <b>Cappriciosa </b>
   Mozzarella, tomato sauce, prosciutto, olives, artichokes
   &amp;amp; mushrooms

   <b>Quatro Staggioni </b>
   Tomato, mozzarella, ham, artichokes &amp;amp; mushrooms
   <b>Margherita </b>
   Fresh tomatoes, fresh mozzarella &amp;amp; basil

   <b>Campagnola </b>
   Tomato, mozzarella, zucchini, eggplant, peppers
   &amp;amp; mushrooms

   <b>Siciliana </b>
   Carciofini, prosciutto, pomidoro, fresh mozzarella
   &amp;amp; whustel
</p>
<p align="center">
   <a href="#Contact">[Contact Us]</a><br/>
   <a href="#Start">[Go to Start]</a>
</p>
</card>
</wml>
```